UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/076,667	02/14/2002	Eric B. Fleegal	MS1-875US	1705

22801 7590 02/09/2005

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER
FOWLKES, ANDRE R

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 02/09/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/076,667	**Applicant(s)** FLEEGAL, ERIC B.
	Examiner Andre R. Fowlkes	**Art Unit** 2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 October 2004*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *13-22 and 38-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *13-22 and 38-46* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 February 2002* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *10/12/04*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. Claims 13-22 and 38-46 are pending.

Election/Restrictions

2. Restriction to one of the following inventions is required under 35 U.S.C. 121:

I. Claims 1-8 and 33-37, drawn to transforming an application programming interface (API) description into a non-markup language source file, including the step of transforming the API definition into a test proxy object code file, classified in class 717, subclass 136.

II. Claims 9-12, drawn to transforming an interface definition into data for a file, including the steps of automatically determining if the interface definition has been changed and re-transforming the interface definition, classified in class 717, subclass 136.

III. Claims 13-22 and 38-46, drawn to transforming an application programming interface description into code for a component object module application programming interface header file, including the step of checking to see whether a declare enumeration construct is to be transformed into a series of manifest constants or into a component object model enumeration declaration, classified in class 717, subclass 136.

IV. Claims 23-32, drawn to transforming an application programming interface description into C or C++ code for a COM application programming interface header file, classified in class 717, subclass 136.

3. The inventions are distinct, each from the other because of the following reasons:

4. Inventions I, II, III and IV are related as combination and subcombination. Inventions in this relationship are distinct if it can be shown that (1) the combination as claimed does not require the particulars of the subcombination as claimed for patentability, and (2) that the subcombination has utility by itself or in other combinations (MPEP § 806.05(c)).

5. In the instant case, invention I as claimed does not require the particulars of invention II, III or IV as claimed because an application programming interface (API) description is transformed into a non-markup language source file without performing all of the techniques claimed in inventions II, III or IV. Invention II as claimed does not require the particulars of invention I, III or IV as claimed because an interface definition is transformed into data for a file without performing all of the techniques claimed in inventions I, III or IV. Invention III as claimed does not require the particulars of invention I, II or IV as claimed because an application programming interface description is transformed into code for a component object module application programming interface header file without performing all of the techniques claimed in inventions I, II or IV. Invention IV as claimed does not require the particulars of invention I, II or III as claimed because an application programming interface description is transformed into C or C++ code for a COM application programming interface header file without performing all of the techniques claimed in inventions I, II or III.

6. Invention I has separate utility such as transforming the API definition into a test proxy object code file. Invention II has separate utility such as automatically determining if the interface definition has been changed and re-transforming the interface definition. Invention III has separate utility such as checking to see whether a declare enumeration construct is to be transformed into a series of manifest constants or into a component object model enumeration declaration and performing the desired transformation. Invention IV has separate utility such as transforming an application programming interface description into C or C++ code for a COM application programming interface header file.

7. During a telephone conversation with Alan Sponseller on 1/4/05 a provisional election was made without traverse to prosecute the invention of group III, claims 13-22 and 38-46. Affirmation of this election must be made by applicant in replying to this Office action. Claims 1-12 and 22-37 are withdrawn from further consideration by the examiner, 37 CFR 1.142(b), as being drawn to a non-elected invention.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 13-22 and 38-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuznetsov, U.S. Patent Application Publication No. 2001/0056504, in view of The Component Object Model Specification, (Comspec), Version 0.9. The paragraph and line numbers of the PGPUB application will be used to cite the reference.

As per claim 13, Kuznetsov discloses **one or more computer readable media having stored thereon a plurality of instructions that, when executed by a transformation engine,** (¶. 7:14-16, "need to provide dynamic conversions, such as ... XML-to-WAP"), **causes the transformation engine to:**

- access a plurality of constructs in an application programming interface description, wherein the description is written in an extensible markup language (XML) format (¶. 5:3-6,"XML allows tags used to define elements of a page or document to be flexibly defined by the developer of the page. Thus (XML) Web pages (i.e. application programming interfaces) can be designed to effectively function like database records"),

- transform each of the plurality of constructs into code for other **application programming interface header file** (¶. 6:4-7, "The tremendous and continuing growth of XML in B2B applications has led to a great number of different XML e-business vocabularies and schemas", and ¶. 7:13-16, "As the diversity of web-connected devices grows, so grows the need to provide dynamic conversion, such as

XML-to-HTML and XML-to-WAP, for e-business applications (i.e. application programming interface header files)").

Kuznetsov doesn't explicitly disclose translation into a **COM** application programming interface header file.

However, Comspec, in an analogous environment, discloses **COM** application programming interface (p. 1:5-7, "This document contains the specification to the Component Object Model (COM), an architecture and supporting infrastructure for building, using, and evolving component software in a robust manner. This specification contains the standard APIs supported by the COM Library").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from an XML API into a COM API header file. The modification would have been obvious because one of ordinary skill in the art would have wanted the flexibliity of converting a recent data encoding format, such as XML, into a the format of an existing technology, such as COM, (Kuznetsov, ¶ 7:13-16).

As per claim 14, the rejection of claim 13 is incorporated and further, Kuznetsov discloses that **the transformation engine comprises a series of instructions executed by one or more processors** (col. 11:2-3, "To transform one XML vocabulary to another, the processor must parse the transform").

As per claim 15, the rejection of claim 13 is incorporated, and further Kuznetsov doesn't explicitly disclose that the plurality of instructions include instructions to:

- check whether a declare enumeration construct is to be transformed into a series of manifest constants or into a component object model enumeration declaration

- transform the enumeration construct into either the series of manifest constants or the component object model enumeration declaration based on the checking

However, Comspec, in an analogous environment, discloses that **the plurality of instructions include instructions to:**

- check whether a declare enumeration construct is to be transformed into a series of manifest constants or into a component object model enumeration declaration (p. 8:30, "this is a manifest constant defined in the header files", and p. 7:1, "enumeration (declaration)", and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM),

- transform the enumeration construct into either the series of manifest constants or the component object model enumeration declaration based on the checking (p. 8:30, "this is a manifest constant defined in the header files", and p. 7:1, "enumeration (declaration)", and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM),

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation, (Kuznetsov, ¶ 12:5-8).

As per claim 16, the rejection of claim 13 is incorporated, and further Kuznetsov doesn't explicitly disclose **instructions to transform a declare enumeration construct into a series of manifest constants** .

However, Comspec, in an analogous environment, discloses **instructions to transform a declare enumeration construct into a series of manifest constants** (p. 8:30, "this is a manifest constant defined in the header files", and p. 7:1, "enumeration (declaration)", and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation, (Kuznetsov, ¶ 12:5-8).

As per claim 17, the rejection of claim 13 is incorporated, and further Kuznetsov doesn't explicitly disclose **instructions to transform a declare enumeration construct into a component object model enumeration declaration** .

However, Comspec, in an analogous environment, discloses **instructions to transform a declare enumeration construct into a component object model enumeration declaration** (p. 8:30, "this is a manifest constant defined in the header files", and p. 7:1, "enumeration (declaration)", and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation, (Kuznetsov, ¶ 12:5-8).

As per claim 18, the rejection of claim 13 is incorporated, and further Kuznetsov doesn't explicitly disclose **instructions to transform a declare function construct into a component object model function declaration**.

However, Comspec, in an analogous environment, discloses **instructions to transform a declare function construct into a component object model function declaration** (p. 8:30, "this is a manifest constant defined in the header files", and p. 7:1, "enumeration (declaration)", and 3:27, "(COM) Function (declaration)", and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation, (Kuznetsov, ¶ 12:5-8).

As per claim 19, the rejection of claim 13 is incorporated, and further Kuznetsov doesn't explicitly disclose **instructions to transform a declare class object construct into a component object model class object ID declaration**.

However, Comspec, in an analogous environment, discloses **instructions to transform a declare class object construct into a component object model class object ID declaration** (p. 8:30, “this is a manifest constant defined in the header files”, and p. 7:1, “enumeration (declaration)”, and p. 3:13, “(COM) object class”, and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation, (Kuznetsov, ¶ 12:5-8).

As per claim 20, the rejection of claim 13 is incorporated, and further Kuznetsov doesn’t explicitly disclose **instructions to transform a declare interface construct into a component object model forward class declaration**.

However, Comspec, in an analogous environment, discloses **instructions to transform a declare interface construct into a component object model forward class declaration** (p. 8:30, “this is a manifest constant defined in the header files”, and p. 7:1, “enumeration (declaration)”, and p. 3:13, “(COM) object class”, and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation, (Kuznetsov, ¶ 12:5-8).

As per claim 21, the rejection of claim 13 is incorporated, and further Kuznetsov doesn't explicitly disclose **instructions to transform a declare data structure construct into a component object model data structure declaration**.

However, Comspec, in an analogous environment, discloses **instructions to transform a declare data structure construct into a component object model data structure declaration** (p. 8:30, "this is a manifest constant defined in the header files", and p. 7:1, "enumeration (declaration)", and p. 4:25, "(Data) structure (declaration)", and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation, (Kuznetsov, ¶ 12:5-8).

As per claim 22, the rejection of claim 13 is incorporated, and further Kuznetsov doesn't explicitly disclose **instructions to transform a declare macro construct into a component object model manifest constant**.

However, Comspec, in an analogous environment, discloses **instructions to transform a declare macro construct into a component object model manifest constant** (p. 8:30, "this is a manifest constant defined in the header files", and p. 7:1, "enumeration (declaration)", and p. 3:13, "(COM) object class", and to perform transformation between XML and COM, the transformation engine maps constructs, constants and declarations in XML to the corresponding constructs, constants and declarations in COM).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Comspec into the system of Kuznetsov to have transformation from the XML constructs, constants and declarations into the corresponding COM constructs, constants and declarations. The modification would have been obvious because one of ordinary skill in the art would have wanted to appropriately map the conventions of each language in order to have a complete and consistent transformation,. (Kuznetsov, ¶ 12:5-8).

As per claims 38-46, this is another computer readable medium version of the claimed medium discussed above, in claims 15 and 18-22, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see (Kuznetsov, ¶ 5:1-7:16 & Comspec, p. 1:5-8:30).

Conclusion

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

ARF

TUAN DAM
SUPERVISORY PATENT EXAMINER